

Case Study 2

Database Design & Prototype

Feedback

CS2.1

PEOPLE: Used to record the festival-goer characteristics when they first enter the festival. You may have identified that there is a potential super/subtype here.

GATE MOVEMENT: This will be used to record the exit and entry into the festival by automatic scanning of the person's wristband.

OBSERVATION: Used to record the researcher's counting observations as well as the start/end of the observation period.

ACTIVITY: This entity will be used to store the different types of activities identified in the scenario. We still need a way of recording these if the researcher observes any people wearing hoodies taking part in these activities. That will be done in the normalisation steps.

ZONE: This will record the details of each zone in the festival area.

There are other entities which are identified from the scenario but which don't have specific instructions to record data for. The RESEARCHER entity will be required for the RR App for example and should also store the Phone and Email for them. This initial design could be enhanced but we will discuss this again in Step 6.

CS2.2

Identify the attributes that will be needed for each entity.

PEOPLE	GATE MOVEMENT
WristbandID Gender AgeGroup HoodyYN JeansYN ThongsYN DressSkirtYN HatYN HairColour GroupSize SuspicionRating HoodyUpYN BulgingPocketsYN Notes	Wristband ID Gate Time InOut

OBSERVATION
Zone
DancingCount
EatingCount
DrinkingCount

ACTIVITY
ActivityType
SuspicionRating

ZONE
ZoneCode
Zone Size

RESEARCHER
ResearcherID
FirstName
LastName
Mobile
Email

Assumptions:

1. We will need to record the StartTime and EndTime of the observations. The test data assumes that this will be in 1-hour blocks.
2. We aren't interested in the address details of the Researcher but we will wish to record the mobile number and email for use in the RR App.
3. MovementType will be one of *Entry*, *In* or *Out*. This will be automatically captured by the security system. *Entry* is used when the festivalgoers enters for the first time.
4. ZoneSize will need to be recorded so that analysis of activity counts can be done by area.

There are some problems with our design. At the moment we will just note the major problem, and ensure it is sorted out in the next steps. There is nowhere to record what activities the HOODY people may be observed doing. We will refine this in the following steps. You may note there is no place to record the time when PEOPLE characteristics are recorded. However, the GATE_MOVEMENT entity holds this information and the observation is made at that time.

CS2.3

PEOPLE: WristbandID is the logical primary key

GATE_MOVEMENT: A composite key could be created using WristbandID and MovementDateTime. Alternatively, a surrogate key could also be used.

OBSERVATION: A surrogate key is used to identify the individual art shows

ACTIVITY: The ActivityName attribute is a candidate for primary key use. However, as the content of this attribute is text, it would be more efficient to add a surrogate key.

ZONE: ZoneID is the logical choice for the primary key as unique values for each zone are already assigned.

RESEARCHER: More than one researcher may have the same last name / first name combination, so a surrogate key of ResearcherID is used.

CS2.4

PEOPLE
Wristband ID
Gender
Age Group
Hoody?
Jeans?
Thongs?
DressSkirt?

HOODY
Group Size
Suspicion Rating
Hoody Up?
Bulging Pockets?
Notes

NON HOODY
Hat?
Hair Colour

What attribute/key would you add to the subtypes to use as the Primary Key?

The WristbandID can be used as the primary key for each of the subtypes.

Which attribute would you use as the discriminator?

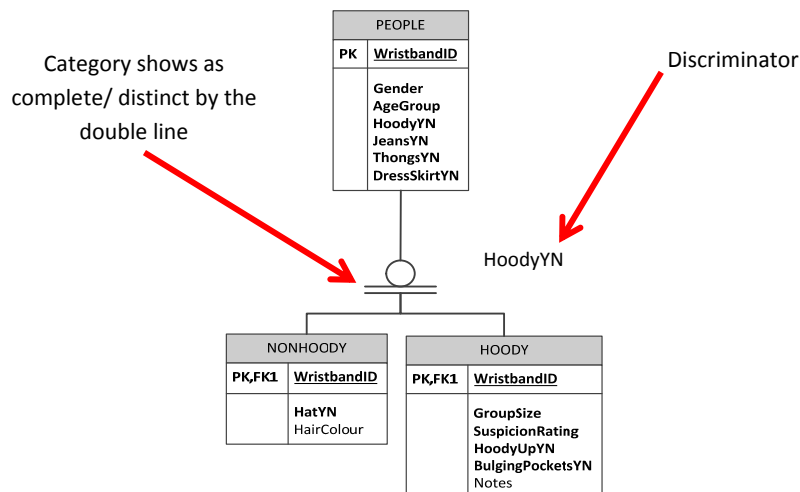
The HoodyYN attribute is used as the discriminator.

Given the fact that a person is either a hoody or a non-hoody, would you say that the category is complete?

The category is 'complete'. Each person must have a row in either of the subtypes, not none or both.

CS2.5

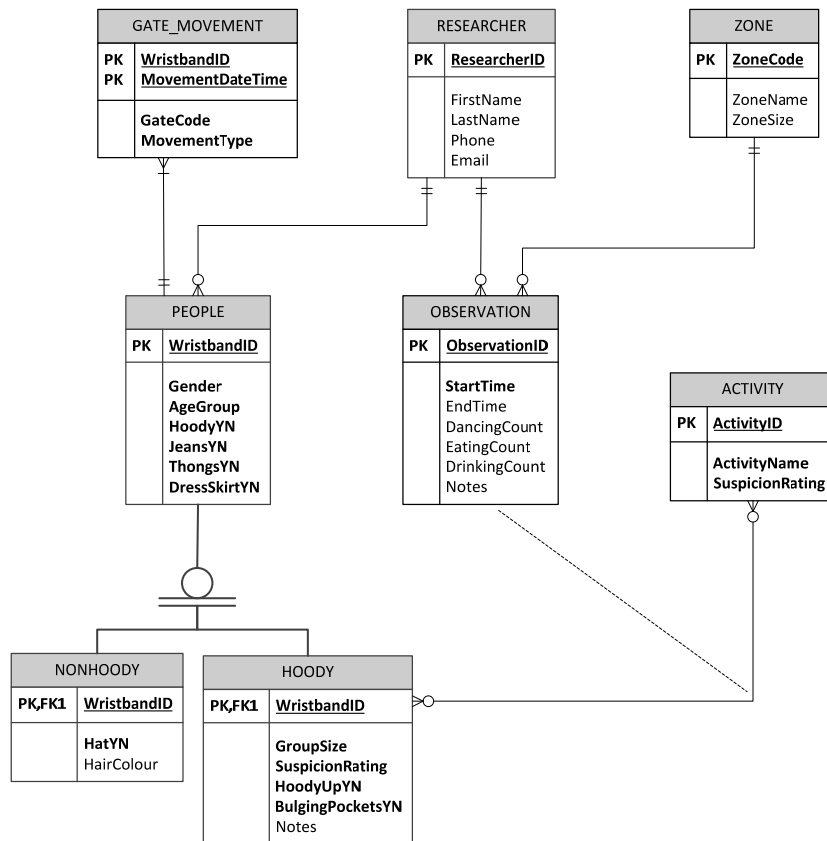
The supertype PEOPLE attributes will apply to all festival-goers. They will be classified as either hoody or non-hoody by the discriminator HoodyYN.



In the above model, the attribute names have also been modified to a consistent format. The spaces and question marks have been removed. Remember that special characters such as a question mark should be avoided if possible. If you have text-only for your attribute names, there is less to consider when using the values when program. For example, if you include a space in your attribute name, you will *always* have to wrap the name in backticks or quotes.

CS2.6

For an explanation of these relationships, please play the screencast (FeedbackCS2_6.mp4) in the Case Study 2 folder.

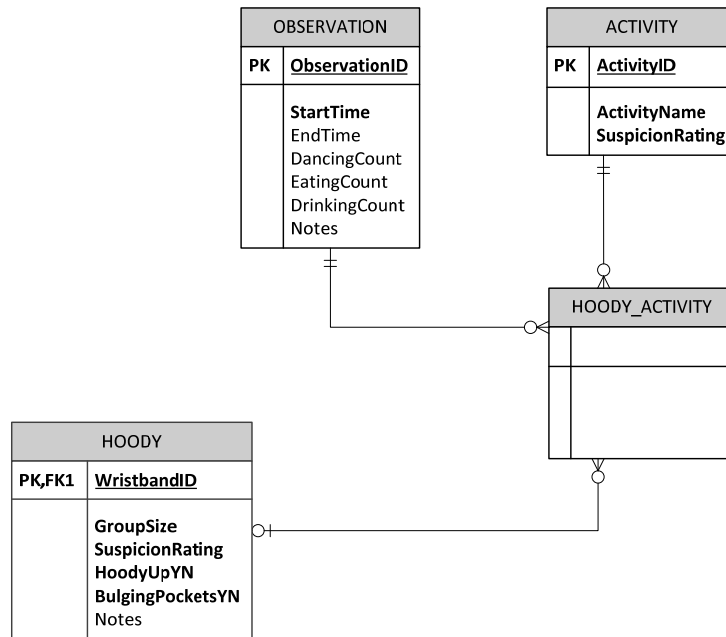


CS2.7

Your model, at this stage, should show all entities and attributes and the super/subtype notation. No other relationships need to be modelled yet.

CS2.8

For an explanation, please play the screencast (FeedbackCS2_8.mp4) in the Case Study 2 folder.



CS2.9

GATE_MOVEMENT(WristbandID, MovementDateTime, GateCode, MovementType)

PEOPLE(WristbandID, Gender, AgeGroup, HoodyYN, JeansYN, ThongsYN, DressSkirtYN)

NONHOODY(WristbandID, HatYN, HairColour)

HOODY(WristbandID, GroupSize, SuspicionRating, HoodyUpYN, BulgingPocketsYN, Notes)

RESEARCHER(ResearcherID, FirstName, LastName, Phone, Email)

OBSERVATION(ObservationID, StartTime, EndTime, DancingCount, EatingCount, DrinkingCount)

ZONE(ZoneCode, ZoneName, ZoneSize)

ACTIVITY(ActivityID, ActivityName, SuspicionRating)

HOODY_ACTIVITY(HoodyActivityID, ObservationID, ActivityID, WristbandID)

HOODY_ACTIVITY is the only additional relation required.

CS2.10

(Add the Foreign Keys)

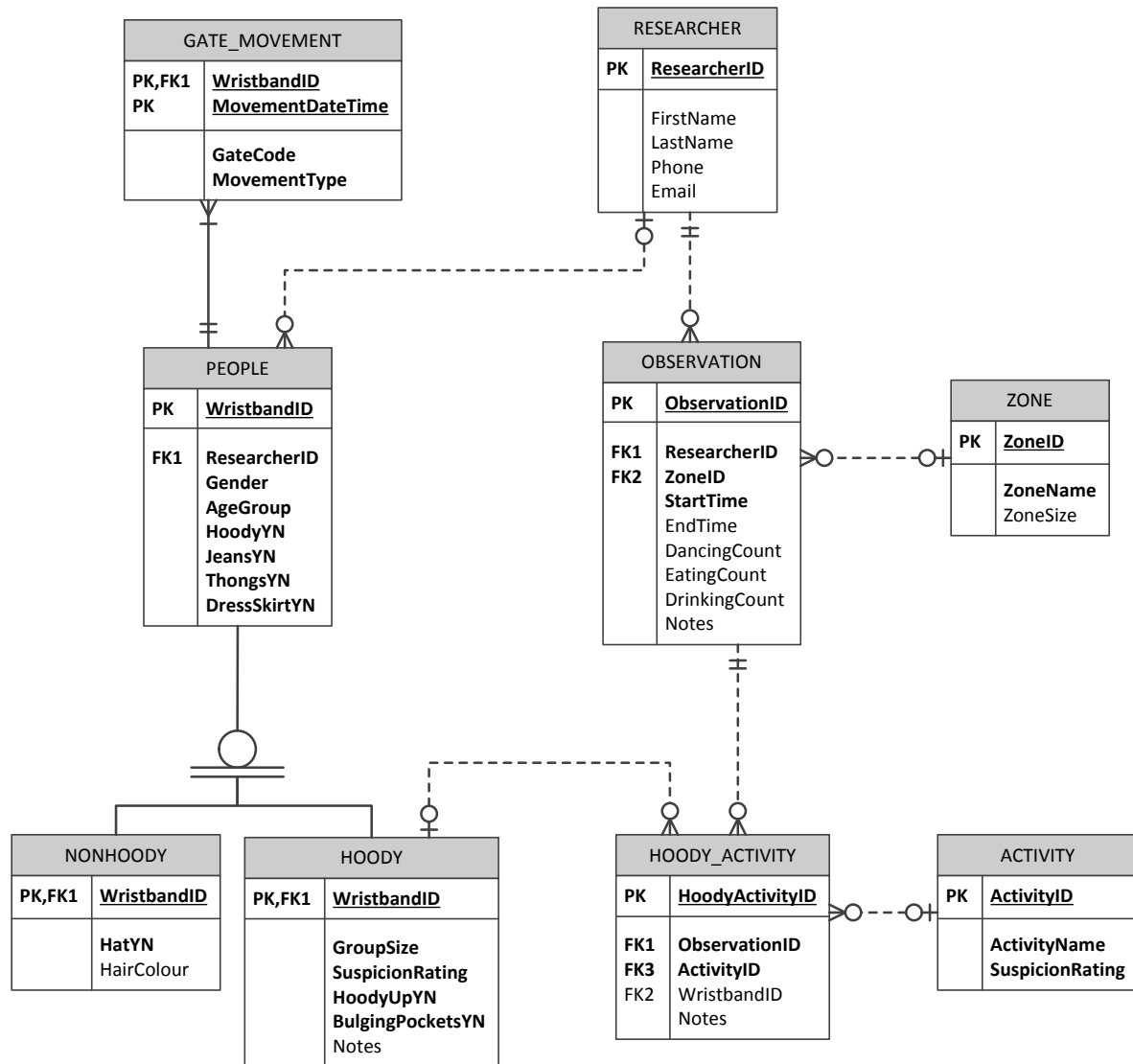
The PEOPLE entity will require the addition of a **ResearcherID** foreign key, to relate it to the RESEARCHER's primary key ResearcherID. This is to record which researcher recorded the data.

The OBSERVATION entity will require the addition of a **ResearcherID** foreign key, to relate it to the RESEARCHER's primary key ResearcherID. It will also require the addition of a **ZoneCode** foreign key, to relate it to the ZONE primary key **ZoneCode**.

RESEARCHER(ResearcherID, FirstName, LastName, Phone, Email)
 PEOPLE (WristbandID, ResearcherID, Gender, AgeGroup, HoodyYN, JeansYN, ThongsYN,
 DressSkirtYN)
 GATE_MOVEMENT(WristbandID, MovementDateTime, GateCode, MovementType)
 NONHOODY(WristbandID, HatYN, HairColour)
 HOODY(WristbandID, GroupSize, SuspicionRating, HoodyUpYN, BulgingPocketsYN, Notes)
 OBSERVATION(ObservationID, ResearcherID, ZoneCode, StartTime, EndTime,
 DancingCount, EatingCount, DrinkingCount)
 ZONE(ZoneCode, ZoneName, ZoneSize)
 ACTIVITY(ActivityID, ActivityName, SuspicionRating)
 HOODY_ACTIVITY(HoodyActivityID, ObservationID, ActivityID, WristbandID)

In this normalisation step, you will notice that most of the normalisation has already occurred. This is partly because the GATE_MOVEMENT entity specifications were provided to you in the specification and partly because you have already resolved the super/subtype relations.

CS2.11



For an explanation, please play the screencast (FeedbackCS2_12.mp4) in the Case Study 2 folder.

CS2.12

PEOPLE							
Attribute	Data Type	Primary Key	Foreign Key	Index Type?	Default?	NULL ?	Auto-increment ?
WristbandID	INT(11)	Yes	No	PRIMARY	No	No	No
Gender	VARCHAR(1)	No	No		No	No	No
AgeGroup	VARCHAR(10)	No	No		No	No	No
HoodyYN	BOOLEAN	No	No		False/0	No	No
JeansYN	BOOLEAN	No	No		False/0	No	No
ThongsYN	BOOLEAN	No	No		False/0	No	No
DressSkirtYN	BOOLEAN	No	No		False/0	No	No

GATE_MOVEMENT							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default?</i>	<i>NULL?</i>	<i>Auto-increment?</i>
<i>WristbandID</i>	<i>INT(11)</i>	<i>Yes</i>	<i>No</i>	<i>PRIMARY</i>	<i>No</i>	<i>No</i>	<i>No</i>
<i>MovementDate Time</i>	<i>DATETIME</i>	<i>Yes</i>	<i>No</i>	<i>PRIMARY</i>	<i>No</i>	<i>No</i>	<i>No</i>
<i>GateCode</i>	<i>VARCHAR(10)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>No</i>	<i>No</i>
<i>MovementType</i>	<i>VARCHAR(5)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>No</i>	<i>No</i>

NONHOODY							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default ?</i>	<i>NULL ?</i>	<i>Auto-increment?</i>
<i>WristbandID</i>	<i>INT(11)</i>	<i>Yes</i>	<i>No</i>	<i>PRIMARY</i>	<i>No</i>	<i>No</i>	<i>No</i>
<i>HatYN</i>	<i>BOOLEAN</i>	<i>No</i>	<i>No</i>		<i>False/0</i>	<i>No</i>	<i>No</i>
<i>HairColour</i>	<i>VARCHAR(15)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>Yes</i>	<i>No</i>

HOODY							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default ?</i>	<i>NULL ?</i>	<i>Auto-increment ?</i>
<u><i>WristbandID</i></u>	<i>INT(11)</i>	<i>Yes</i>	<i>No</i>	<i>PRIMARY</i>	<i>No</i>	<i>No</i>	<i>No</i>
<i>GroupSize</i>	<i>VARCHAR(15)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>Yes</i>	<i>No</i>
<i>SuspicionRating</i>	<i>INT(1)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>Yes</i>	<i>No</i>
<i>HoodyUpYN</i>	<i>BOOLEAN</i>	<i>No</i>	<i>No</i>		<i>False/0</i>	<i>No</i>	<i>No</i>
<i>BulgingPocketsYN</i>	<i>BOOLEAN</i>	<i>No</i>	<i>No</i>		<i>False/0</i>	<i>No</i>	<i>No</i>
<i>Notes</i>	<i>VARCHAR(255)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>Yes</i>	<i>No</i>

RESEARCHER							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default ?</i>	<i>NULL ?</i>	<i>Auto-increment?</i>
<u><i>ResearcherID</i></u>	<i>INT(11)</i>	<i>Yes</i>	<i>No</i>	<i>PRIMARY</i>	<i>No</i>	<i>No</i>	<i>Yes</i>
<i>FirstName</i>	<i>VARCHAR(30)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>No</i>	<i>No</i>
<i>LastName</i>	<i>VARCHAR(30)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>No</i>	<i>No</i>
<i>Phone</i>	<i>VARCHAR(15)</i>	<i>No</i>	<i>No</i>		<i>No</i>	<i>No</i>	<i>No</i>
<i>Email</i>	<i>VARCHAR(255)</i>	<i>No</i>	<i>No</i>	<i>UNIQUE</i>	<i>No</i>	<i>No</i>	<i>No</i>

OBSERVATION							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default?</i>	<i>NULL?</i>	<i>No</i>
<u>ObservationID</u>	INT(11)	Yes	No	PRIMARY	No	No	Yes
ResearcherID	INT(11)	No	Yes	INDEX	No	No	No
ZoneID	VARCHAR(4)	No	Yes	INDEX	No	No	No
StartTime	DATETIME	No	No	INDEX	No	No	No
EndTime	DATETIME	No	No	INDEX	No	Yes	No
DancingCount	INT(5)	No	No		0	No	No
EatingCount	INT(5)	No	No		0	No	No
DrinkingCount	INT(5)	No	No		0	No	No

ZONE							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default ?</i>	<i>NUL L?</i>	<i>Auto-increment?</i>
ZoneID	VARCHAR(4)	Yes	No	PRIMARY	No	No	Yes
ZoneName	VARCHAR(20)	No	No		No	No	No
ZoneSize	DECIMAL(3,2)	No	No		0	No	No

ACTIVITY							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default?</i>	<i>NULL?</i>	<i>Auto-increment?</i>
ActivityID	INT(11)	Yes	No	PRIMARY	No	No	Yes
ActivityName	VARCHAR(30)	No	No		No	No	No
SuspicionRating	INT(1)	No	No		0	No	No

HOODY_ACTIVITY							
<i>Attribute</i>	<i>Data Type</i>	<i>Primary Key</i>	<i>Foreign Key</i>	<i>Index Type?</i>	<i>Default?</i>	<i>NULL?</i>	<i>Auto-increment?</i>
HoodyActivityID	INT(11)	Yes	No	PRIMARY	No	No	Yes
ObservationID	INT(11)	No	Yes	INDEX	No	No	No
ActivityID	INT(11)	No	Yes	INDEX	No	No	No
WristbandID	INT(11)	No	Yes	INDEX	No	No	No
Notes	VARCHAR(255)	No	No		No	Yes	No

CS2.13

The ACTIVITY table stores the list of activities that the researchers are to observe. The UNIQUE KEY has been used to ensure that the ActivityName is unique. Some applications may give the user the option of adding to the list. This UNIQUE KEY setting will ensure no duplicates.

```
CREATE TABLE ACTIVITY (  
    ActivityID          int(11)          NOT NULL AUTO_INCREMENT,  
    ActivityName        varchar(30)       NOT NULL,  
    SuspicionRating     int(1)           NOT NULL,  
    PRIMARY KEY         (ActivityID),  
    UNIQUE KEY          ActivityName     (ActivityName)  
);
```

The GATE_MOVEMENT table has a composite primary key

```
CREATE TABLE IF NOT EXISTS `GATE_MOVEMENT` (  
    WristbandID        int(11)          NOT NULL,  
    Movement           DateTime datetime NOT NULL,  
    GateCode           varchar(10)       NOT NULL,  
    MovementType       varchar(5)       NOT NULL,  
    PRIMARY KEY         (WristbandID, MovementDateTime)  
);
```

```
CREATE TABLE HOODY (  
    WristbandID        int(11)          NOT NULL,  
    GroupSize          int(2)           DEFAULT NULL,  
    SuspicionRating     int(1)           NOT NULL DEFAULT '0',  
    HoodyUpYN          tinyint(1)       NOT NULL DEFAULT '0',  
    BulgingPocketsYN    tinyint(1)       NOT NULL DEFAULT '0',  
    Notes              varchar(255)      DEFAULT NULL,  
    PRIMARY KEY         (WristbandID)  
);
```

```
CREATE TABLE HOODY_ACTIVITY (  
    HoodyActivityID    int(11)          NOT NULL AUTO_INCREMENT,  
    ObservationID       int(11)          NOT NULL,  
    ActivityID          int(11)          NOT NULL,  
    WristbandID        int(11)          NOT NULL,  
    Notes              varchar(255)      DEFAULT NULL,  
    PRIMARY KEY         (HoodyActivityID),  
    KEY                ObservationID (ObservationID),  
    KEY                ActivityID (ActivityID),  
    KEY                WristbandID (WristbandID)  
);
```

```

CREATE TABLE NONHOODY (
    WristbandID          int(11)          NOT NULL,
    HatYN tiny           int(1)           NOT NULL DEFAULT '0',
    HairColour           varchar(15)       NOT NULL,
    PRIMARY KEY          (WristbandID)
);

CREATE TABLE OBSERVATION (
    ObservationID        int(11)          NOT NULL AUTO_INCREMENT,
    ResearcherID         int(11)          NOT NULL,
    ZoneID              varchar(4)        NOT NULL,
    StartTime            datetime         NOT NULL,
    EndTime              datetime         DEFAULT NULL,
    DancingCount         int(5)           NOT NULL,
    EatingCount          int(5)           NOT NULL,
    DrinkingCount        int(5)           NOT NULL,
    PRIMARY KEY          (ObservationID),
    KEY                  ResearcherID     (ResearcherID),
    KEY                  ZoneID           (ZoneID)
);

CREATE TABLE PEOPLE (
    WristbandID          int(11)          NOT NULL,
    ResearcherID         int(11)          NOT NULL,
    Gender              varchar(1)        NOT NULL,
    AgeGroup            varchar(15)       NOT NULL,
    HoodyYN tiny         int(1)           NOT NULL,
    JeansYN tiny         int(1)           NOT NULL,
    ThongsYN tiny        int(1)           NOT NULL,
    DressSkirtYN tiny    int(1)           NOT NULL,
    PRIMARY KEY          (WristbandID),
    KEY                  ResearcherID     (ResearcherID)
);

CREATE TABLE RESEARCHER (
    ResearcherID        int(11)          NOT NULL AUTO_INCREMENT,
    FirstName            varchar(30)       NOT NULL,
    LastName             varchar(30)       NOT NULL,
    Phone                varchar(15)       NOT NULL,
    Email                varchar(255)      NOT NULL,
    PRIMARY KEY          (ResearcherID)
);

CREATE TABLE ZONE (
    ZoneID              varchar(4)        NOT NULL,
    ZoneName            varchar(20)       NOT NULL,
    ZoneSize             decimal(3,2)     NOT NULL,
    PRIMARY KEY          (ZoneID)
);

```

CS2.14

For the table HOODY, we also need to add the constraints

```
ALTER TABLE HOODY
ADD CONSTRAINT HOODY_ibfk_1 FOREIGN KEY (WristbandID)
REFERENCES PEOPLE (WristbandID);
```

For the table HOODY_ACTIVITY:

```
ALTER TABLE HOODY_ACTIVITY
ADD CONSTRAINT HOODY_ACTIVITY_ibfk_1 FOREIGN KEY (ActivityID) REFERENCES
ACTIVITY (ActivityID),
ADD CONSTRAINT HOODY_ACTIVITY_ibfk_2 FOREIGN KEY (WristbandID) REFERENCES
HOODY (WristbandID),
ADD CONSTRAINT HOODY_ACTIVITY_ibfk_3 FOREIGN KEY (ObservationID) REFERENCES
OBSERVATION (ObservationID);
```

For the table NONHOODY

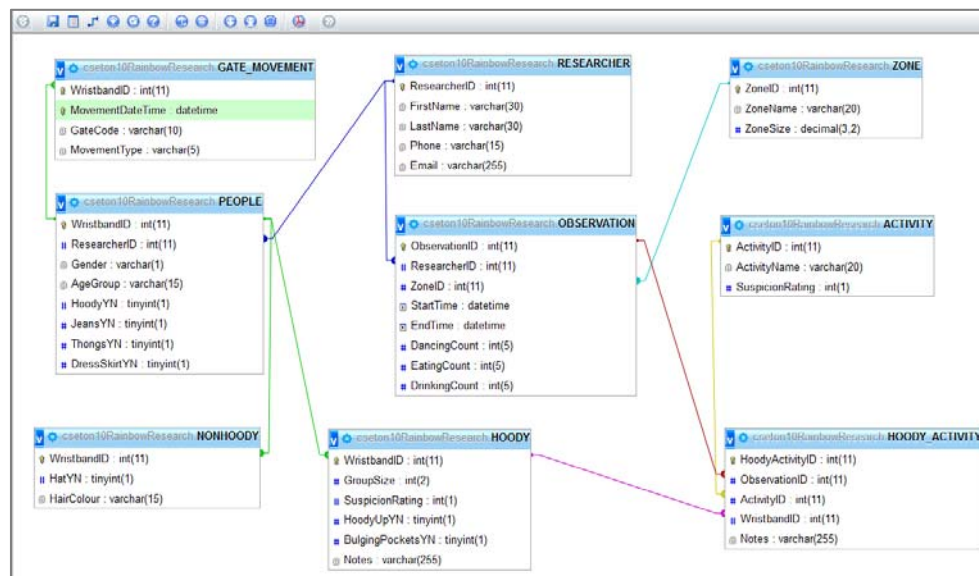
```
ALTER TABLE NONHOODY
ADD CONSTRAINT NONHOODY_ibfk_1 FOREIGN KEY (WristbandID) REFERENCES
PEOPLE (WristbandID);
```

For the table OBSERVATION:

```
ALTER TABLE OBSERVATION
ADD CONSTRAINT OBSERVATION_ibfk_2 FOREIGN KEY (ZoneID) REFERENCES
ZONE (ZoneID) ON UPDATE CASCADE,
ADD CONSTRAINT OBSERVATION_ibfk_1 FOREIGN KEY (ResearcherID) REFERENCES
RESEARCHER (ResearcherID);
```

For the table PEOPLE:

```
ALTER TABLE PEOPLE
ADD CONSTRAINT PEOPLE_ibfk_1 FOREIGN KEY (ResearcherID) REFERENCES
RESEARCHER (ResearcherID);
```



CS2.15

The following INSERT statements will add data into the tables RESEARCHER, ZONE and ACTIVITY. Note that with MySQL, you will need to specify the column names apart from the auto-incrementing key, to use the key correctly, and all non-numerical data should be quoted.

There are various tools for creating test data available on the net. One is "Spawner" which allows you to specify the fields and what you would like to have included.

The address for Spawner is <http://spawner.sourceforge.net/>

RESEARCHER:

INSERT INTO RESEARCHER (FirstName, LastName, Phone, Email) VALUES

```
('Jane', 'Howard', '0499 111 566', 'jane.howard@rr.com.au'),
('Peter', 'Numoto', '0499 222 566', 'peter.howard@rr.com.au'),
('Jane', 'Nestor', '0499 111 333', 'jane.howard@rr.com.au'),
('Rick', 'Cameron', '0499 444 566', 'rick.howard@rr.com.au'),
('Heather', 'Mason', '0499 222 222', 'heather.mason@rr.com.au'),
('Jasmine', 'Caruthers', '0499 111 322', 'jasmine.caruthers@rr.com.au'),
('Jane', 'Brown', '0499 333 333', 'jane.brown@rr.com.au'),
('George', 'Chan', '0499 111 322', 'george.chan@rr.com.au'),
('Singh', 'Jackson', '0499 543 566', 'singh.jackson@rr.com.au'),
('Paul', 'Wong', '0499 444 566', 'paul.wong@rr.com.au'),
('Jane', 'Edmonds', '0499 222 566', 'jane.edmonds@rr.com.au');
```

ZONE:

INSERT INTO ZONE (ZoneID, ZoneName, ZoneSize) VALUES

```
('MJ01', 'Mojo Tent 1', 0.5),
('MJ02', 'Mojo Tent 2', 0.4),
('MJ03', 'Mojo North', 1.1),
('MJ04', 'Mojo Tent South', 0.8),
('CR01', 'Crossroads Tent 1', 1),
('CR02', 'Crossroads Tent 2', 0.5),
('CR03', 'Crossroads South', 0.3),
('JM01', 'Jambalaya Tent', 0.4),
('TT01', 'Techno Tent 1', 1.5),
('FD01', 'Food Area 1', 1.5),
('FD02', 'Food Area 2', 1.3);
```

INSERT INTO ACTIVITY (ActivityName, SuspicionRating) VALUES

```
('standing near the toilet', 5),
('approaching strangers', 8),
('crazy behaviour', 7),
('falling over', 5),
('aggression verbal', 5),
('aggression physical', 5);
```

CS2.16

INSERT INTO GATE_MOVEMENT

```
(WristbandID , MovementDateTime , GateCode , MovementType ) VALUES
(1345000123, '2013-04-12 13:01:00', 'Main', 'Entry'),
(1345000112, '2013-04-12 13:01:02', 'Main', 'Entry'),
(1345000113, '2013-04-12 13:01:03', 'Main', 'Entry'),
(1345000114, '2013-04-12 13:02:04', 'Main', 'Entry'),
(1345001233, '2013-04-12 13:02:22', 'Main', 'Entry'),
(1345009777, '2013-04-12 13:02:47', 'Main', 'Entry'),
(1345007533, '2013-04-12 13:03:00', 'Main', 'Entry'),
(1345000322, '2013-04-12 13:03:33', 'Main', 'Entry'),
(1345000546, '2013-04-12 13:03:56', 'Main', 'Entry'),
(1345004266, '2013-04-12 13:04:12', 'Main', 'Entry'),
(1345007833, '2013-04-12 13:04:14', 'Main', 'Entry'),
(1345000555, '2013-04-12 13:06:22', 'Main', 'Entry'),
(1345000732, '2013-04-12 13:08:22', 'Main', 'Entry'),
(1345000001, '2013-04-12 13:08:44', 'Main', 'Entry'),
(1345000123, '2013-04-12 14:01:00', 'Main', 'Out'),
(1345000123, '2013-04-12 14:09:00', 'Main', 'In'),
(1345000123, '2013-04-12 14:22:00', 'South', 'Out'),
(1345000123, '2013-04-12 15:12:00', 'South', 'In'),
(1345000123, '2013-04-12 15:22:00', 'South', 'Out'),
(1345000123, '2013-04-12 15:27:00', 'South', 'In'),
(1345007833, '2013-04-12 15:04:14', 'East', 'Out'),
(1345007833, '2013-04-12 15:34:11', 'East', 'In');
```

INSERT INTO OBSERVATION (ObservationID, ResearcherID, ZoneID, StartTime, EndTime, DancingCount, EatingCount, DrinkingCount) VALUES

```
(1, '1', 'CR01', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '1543', '126', '157'),
(2, '2', 'CR02', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '1003', '55', '455'),
(3, '9', 'CR03', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '973', '120', '35'),
(4, '5', 'FD01', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '12', '3450', '2157'),
(5, '6', 'FD02', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '55', '2050', '1157'),
(6, '7', 'JM01', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '920', '13', '87'),
(7, '8', 'TT01', '2013-04-12 13:00:00', '2013-04-12 14:00:00', '3433', '10', '247'),
(8, '1', 'CR01', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '1113', '126', '127'),
(9, '2', 'CR02', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '1003', '55', '435'),
(10, '9', 'CR03', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '799', '160', '225'),
(11, '5', 'FD01', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '33', '3850', '1857'),
(12, '6', 'FD02', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '55', '2850', '957'),
(13, '7', 'JM01', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '970', '13', '7'),
(14, '8', 'TT01', '2013-04-12 14:00:00', '2013-04-12 15:00:00', '3233', '10', '57');
```

```

INSERT INTO HOODY_ACTIVITY (HoodyActivityID, ObservationID, ActivityID,
WristbandID, Notes) VALUES
    (NULL, '2', '2', '1345000123', NULL),
    (NULL, '2', '6', NULL, NULL),
    (NULL, '2', '2', '1345000123', NULL),
    (NULL, '3', '3', '1345000123', NULL),
    (NULL, '3', '7', '1345000123', NULL),
    (NULL, '3', '4', '1345000123', NULL),
    (NULL, '5', '3', NULL, NULL),
    (NULL, '8', '7', NULL, NULL),
    (NULL, '12', '4', '1345007833', NULL);

```

```

INSERT INTO `PEOPLE` (`WristbandID`, `ResearcherID`, `Gender`, `AgeGroup`, `HoodyYN`,
`JeansYN`, `ThongsYN`, `DressSkirtYN`) VALUES
    (1345000001, 4, 'm', '<20', 0, 0, 0, 0),
    (1345000112, 3, 'm', '<20', 0, 0, 0, 0),
    (1345000113, 3, 'f', '<20', 0, 0, 0, 1),
    (1345000114, 3, 'f', '20-30', 0, 0, 0, 1),
    (1345000123, 3, 'm', '<20', 1, 1, 1, 0),
    (1345000322, 3, 'f', '20-30', 0, 1, 0, 0),
    (1345000546, 3, 'f', '20-30', 0, 1, 0, 0),
    (1345000555, 3, 'f', '20-30', 0, 0, 0, 1),
    (1345000732, 3, 'f', '<20', 0, 0, 0, 1),
    (1345001233, 3, 'f', '<20', 0, 1, 0, 0),
    (1345004266, 3, 'm', '41-50', 0, 1, 0, 0),
    (1345007533, 3, 'm', '20-30', 1, 0, 0, 0),
    (1345007833, 3, 'm', '20-30', 1, 1, 1, 0),
    (1345009777, 3, 'f', '<20', 0, 0, 0, 1);

```

```

INSERT INTO HOODY (WristbandID, GroupSize, SuspicionRating, HoodyUpYN,
BulgingPocketsYN, Notes) VALUES
    ('1345000123', '6', '8', '1', '1', 'Hmm...'),
    ('1345007533', '2', '0', '0', '0', null),
    ('1345007833', '3', '4', '0', '0', 'weird');

```

```

INSERT INTO NONHOODY (WristbandID, HatYN, HairColour) VALUES
    (1345000001, 0, 'red'),
    (1345000112, 0, 'light blonde'),
    (1345000113, 0, 'dark'),
    (1345000114, 0, 'dark'),
    (1345000322, 1, 'light blonde'),
    (1345000546, 0, 'light blonde'),
    (1345000555, 0, 'light blonde'),
    (1345000732, 1, 'brown'),
    (1345001233, 0, 'brown'),
    (1345004266, 0, 'dark blonde'),
    (1345009777, 0, 'dark blonde');

```

CS2.17

- View to show all information about people with hoodies. The result should show like this:

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	GroupSize	SuspicionRating	HoodyUpYN	BulgingPocketsYN	Notes
1345000123		3 m	<20	1	1	1	0	6	8	1	1	Hmm...
1345007533		3 m	20-30	1	0	0	0	2	0	0	0	NULL
1345007833		3 m	20-30	1	1	1	0	3	4	0	0	weird

CREATE OR REPLACE VIEW People_Hoody AS

```
SELECT PEOPLE.*, HOODY.GroupSize, HOODY.SuspicionRating, HOODY.HoodyUpYN
, HOODY.BulgingPocketsYN , HOODY.Notes FROM PEOPLE
LEFT JOIN HOODY ON (PEOPLE.WristbandID = HOODY.WristbandID)
WHERE HoodyYN = 1;
```

- View to show all information about people without hoodies

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	HatYN	HairColour
1345000001	4	m	<20	0	0	0	0	0	red
1345000112	3	m	<20	0	0	0	0	0	light blonde
1345000113	3	f	<20	0	0	0	1	0	dark
1345000114	3	f	20-30	0	0	0	1	0	dark
1345000322	3	f	20-30	0	1	0	0	1	light blonde

CREATE OR REPLACE VIEW People_NonHoody AS

```
SELECT PEOPLE.*, NONHOODY.HatYN, NONHOODY.HairColour
FROM PEOPLE
LEFT JOIN NONHOODY ON (PEOPLE.WristbandID = NONHOODY.WristbandID)
WHERE HoodyYN = 0;
```

- View to show all information about people with or without hoodies

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	HatYN	HairColour	GroupSize	SuspicionRating	HoodyUpYN	BulgingPocketsYN
1345000001	4	m	<20	0	0	0	0	0	red	NULL	NULL	NULL	NULL
1345000112	3	m	<20	0	0	0	0	0	light blonde	NULL	NULL	NULL	NULL
1345000113	3	f	<20	0	0	0	1	0	dark	NULL	NULL	NULL	NULL
1345000114	3	f	20-30	0	0	0	1	0	dark	NULL	NULL	NULL	NULL
1345000123	3	m	<20	1	1	1	0	NULL	NULL	6	8	1	1
1345000322	3	f	20-30	0	1	0	0	1	light blonde	NULL	NULL	NULL	NULL

CREATE OR REPLACE VIEW People_All AS

```
SELECT PEOPLE.*, NONHOODY.HatYN, NONHOODY.HairColour, HOODY.GroupSize,
HOODY.SuspicionRating, HOODY.HoodyUpYN , HOODY.BulgingPocketsYN ,
HOODY.Notes FROM PEOPLE
LEFT JOIN HOODY ON (PEOPLE.WristbandID = HOODY.WristbandID)
LEFT JOIN NONHOODY ON (PEOPLE.WristbandID = NONHOODY.WristbandID);
```

- View to show all information about observations with researcher name and ZoneName

ObservationID	ResearcherID	ZoneID	StartTime	EndTime	DancingCount	EatingCount	DrinkingCount	ZoneName	Researcher	ZoneSize
3	9	CR03	2013-04-12 13:00:00	2013-04-12 14:00:00	973	120	35	Crossroads South	Jackson Singh	0.30
1	1	CR01	2013-04-12 13:00:00	2013-04-12 14:00:00	1543	126	157	Crossroads Tent 1	Howard Jane	1.00
2	2	CR02	2013-04-12 13:00:00	2013-04-12 14:00:00	1003	55	455	Crossroads Tent 2	Numoto Peter	0.50
4	5	FD01	2013-04-12 13:00:00	2013-04-12 14:00:00	12	3450	2157	Food Area 1	Mason Heather	1.50
5	6	FD02	2013-04-12 13:00:00	2013-04-12 14:00:00	55	2050	1157	Food Area 2	Caruthers Jasmine	1.30
6	7	JM01	2013-04-12 13:00:00	2013-04-12 14:00:00	920	13	87	Jambalaya Tent	Brown Jane	0.40
7	8	TT01	2013-04-12 13:00:00	2013-04-12 14:00:00	3433	10	247	Techno Tent 1	Chan George	1.50

CREATE OR REPLACE VIEW ObservationDetails AS

```
SELECT OBSERVATION.*, ZoneName, concat(LastName, " ", FirstName) AS Researcher,
ZoneSize
FROM OBSERVATION
JOIN RESEARCHER ON ( OBSERVATION.ResearcherID = RESEARCHER.ResearcherID )
JOIN ZONE ON ( OBSERVATION.ZoneID = ZONE.ZoneID )
ORDER BY StartTime, ZoneName;
```